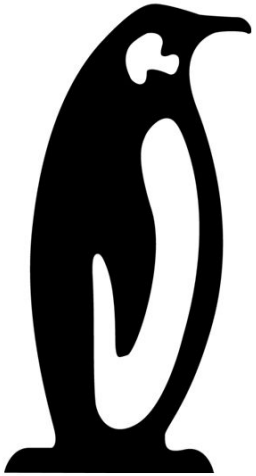

Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024

Multi-Gen LRU Updates

Multi-Gen LRU (MGLRU)

- Alternative LRU implementation for page aging & reclaim
- Key features
  - Organize pages into multi-tier, multi-generational LRUs based on access recency
    - Each LRU generation has a birth timestamp, which allows recency comparison across LRUs
  - Optimize page generation updates via efficient forward page table scans
  - Simplify reclaim between anon and file LRUs using a refault/eviction based PID controller
- Current status
  - Merged into 6.1 two years ago
  - Enabled by default in major distros
    - Debian, Fedora, Ubuntu, Android, ChromeOS

Adopting MGLRU in Google Production

- Historically Google production runs kstaled [1] for time-based proactive aging & reclaim
  - kstaled scans in physical page order with rmap lookups, thus it's cache unfriendly
- As a first step, added small wrappers so MGLRU can emulate kstaled-like time-based aging and reclaim
  - Periodic aging and per-job working set reporting
  - Page age based reclaim
- This is now running in a portion of Google production
  - Achieved same memory savings as kstaled, with 5x less CPU overhead
- Extended MGLRU with efficient KVM secondary MMU aging [2]
  - Scan secondary MMU during aging (vs at eviction only) for more accurate working set reporting
  - Lockless scan can reduce KVM MMU lock contention by ~85% [3]

[1] https://lore.kernel.org/lkml/1317170947-17074-1-git-send-email-walken@google.com/
[2] https://lore.kernel.org/all/20240724011037.3671523-11-jthoughton@google.com/
[3] https://lore.kernel.org/r/20230526234435.662652-1-yuzhao@google.com

Known MGLRU Problems

- OOM kills happen if writeback falls behind
  - There is a fix (“wake up flushers conditionally to avoid cgroup OOM”) posted on the mailing list
  - Further work needed in this area:
    - Smarter decisions about whether to wake up
    - Smarter throttling
- MGLRU has kconfig constraints to ensure enough spare page flag bits
  - `depends on 64BIT || !SPARSEMEM || SPARSEMEM_VMEMMAP`
  - MGLRU needs 3 page flag bits at minimum
  - Proposal: Store `PG_active`, `PG_unevictable` (and possibly `PG_swapback`) in `folio->lru` low bits
- Solicit feedback on other problems

New Use Cases

- Workingset reporting [1] for proactive reclaim and aging
  - Generalization of active/inactive counts to arbitrary time intervals
  - Per-cgroup, per-NUMA node, per swapped-backed/file-backed
  - Working on benchmarks and ballooning use cases as well
- Different heuristics for different types of workloads
  - Aging / reclaim policies can be implemented in BPF, allowing MGLRU to be customized
  - Workloads that aren't well supported by the out-of-the-box MGLRU today can use it
  - Especially relevant for hyperscaler adoption of MGLRU

[1] https://lore.kernel.org/linux-mm/20240813165619.748102-1-yuanchu@google.com/